

Sequenz-Alignment

Jan Schäfer



WS 2006/07

Betreuer: Prof. Dr. Klaus Quibeldey-Cirkel

Überblick

- Einführung
- Grundlagen
- Wann ist das Merkmal der Ähnlichkeit erfüllt ?
- Sequenz-Alignments in Dotplot
- Fazit
- Quellen

Einführung



Ursprung

Bei dem Sequenz-Alignment (engl. Alignment – Abgleich, Anordnung) werden Zeichenketten auf ähnliche Merkmale hin untersucht.

Was ist Ähnlichkeit ?



Was ist Ähnlichkeit ?

Der Grad der Ähnlichkeit bemisst sich nach dem Verhältnis der gemeinsamen zu den unterscheidenden Eigenschaften.

Was ist Ähnlichkeit ?

Der Grad der Ähnlichkeit bemisst sich nach dem Verhältnis der gemeinsamen zu den unterscheidenden Eigenschaften.

Sind keine unterscheidenden Eigenschaften festzustellen, spricht man von Gleichheit oder Identität.

Problemstellung aus der Informatik

Um Gleichheit zwischen zwei Worten festzustellen, kann man in der Informatik den direkten (binären) Vergleich durchführen.

Problemstellung aus der Informatik



Um eine Ähnlichkeit in Worten zu entdecken, gibt es jedoch keine grundlegende Operation.

Verwendung

- Bioinformatik: Erkennung von ähnlicher Funktion und Struktur bei Genen und Proteinen

Verwendung

- Bioinformatik: Erkennung von ähnlicher Funktion und Struktur bei Genen und Proteinen
- Erkennung von Plagiaten

Verwendung

- Bioinformatik: Erkennung von ähnlicher Funktion und Struktur bei Genen und Proteinen
- Erkennung von Plagiaten
- biometrische Verfahren

Verwendung

- Bioinformatik: Erkennung von ähnlicher Funktion und Struktur bei Genen und Proteinen
- Erkennung von Plagiaten
- biometrische Verfahren
- ...

Grundlagen

The slide features a light green background. On the left side, there is a dark green vertical bar. A white rounded rectangle is positioned in the upper left, containing the title. A thick dark blue horizontal bar spans the width of the slide below the title.

Sequenz-Alignment Typen

- Paarweises Alignment
- Multiples Alignment
- Globales Alignment
 - *Semi-Globales Alignment*
- Lokales Alignment

Paarweises Alignment

- Methode, um zwei (Zeichen-)Sequenzen zu vergleichen.

Paarweises Alignment

- Methode, um zwei (Zeichen-)Sequenzen zu vergleichen.
- Als Resultat des paarweisen Alignments erhält man zwei neue Sequenzen, die man als aligniert bezeichnet.

Beispiel: Paarweises Alignment

Eingabe:

```
a="gapa"  
b="gpa"
```

Beispiel: Paarweises Alignment

Eingabe:

```
a="gapa"  
b="gpa"
```

Resultat:

```
a="gapa"  
b="g-pa"
```

Multiples Alignment



Bei der Methode des multiplen Alignments werden mindestens drei Sequenzen aligniert.

Beispiel: Multiples Alignment

Eingabe:

a=N A F L S

b=N A F S

c=N A K Y L S

d= N A Y L S

Beispiel: Multiples Alignment

Eingabe:

a=N A F L S

b=N A F S

c=N A K Y L S

d= N A Y L S

Resultat:

a= N A - F L S

b= N A - F - S

c= N A K Y L S

d= N A - Y L S

Globales Alignment

- bei dem globalen Alignment werden alle Zeichen der Sequenzen analysiert

Globales Alignment

- bei dem globalen Alignment werden alle Zeichen der Sequenzen analysiert
- Die zu untersuchenden Sequenzen sollten jedoch ähnliche Längen aufweisen

Semi-globales Alignment

- Komplette Sequenzen werden aligniert

Semi-globales Alignment

- Komplette Sequenzen werden aligniert
- Fehlende Zeichen die am Anfang oder am Ende eines Wortes auftreten, werden nicht berücksichtigt

Lokales Alignment

- Bei dem lokalen Alignment werden Teilbereiche mit Übereinstimmungen gesucht

Lokales Alignment

- Bei dem lokalen Alignment werden Teilbereiche mit Übereinstimmungen gesucht.
- Der Teilbereich mit der größten Übereinstimmung wird als Ergebnis zurückgegeben

Lokales Alignment

- Bei dem lokalen Alignment werden Teilbereiche mit Übereinstimmungen gesucht.
- Der Teilbereich mit der größten Übereinstimmung wird als Ergebnis zurückgegeben.
- Unterschiedliche Längen der Sequenzen können ohne Probleme verglichen werden

Beispiel: Lokales Alignment

Eingabe:

a="gap"

b="gpa"

Beispiel: Lokales Alignment

Eingabe:

a="gapa"

b="gpa"

Resultat:

a="pa"

b="pa"

Bewertung

Um eine Aussagekraft zu erhalten, müssen Resultate wie „g-pa“ einer Bewertung unterzogen werden.

Definition der Kostenfunktion

- In der Regel hat die Kostenfunktion 3 Parameter

Definition der Kostenfunktion

- In der Regel hat die Kostenfunktion 3 Parameter
- Es müssen Bewertungen für
 - Übereinstimmungen (matches),
 - Abweichungen (dismatches) sowie
 - Lücken (Gaps) definiert werden.

Beispiel: Kostenfunktion

match = 1

dismatch = -1

gap = -2

Beispiel: Kostenfunktion

Eingabe:

k="gapalala"

l= "gapalul"

Resultat:

k="gapalala"

l= "gapalul-"

match = 1

dismatch = -1

gap = -2

Beispiel: Kostenfunktion

Eingabe:

k="gapalala"

l="gapalul"

match = 1

dismatch = -1

gap = -2

Resultat:

k="gapalala"

l="gapalul-"

Ergebnis der Kostenfunktion:

k = g a p a l a l a

l = g a p a l u l -

$$1+1+1+1+1-1+1-2 = 3$$

**Wann ist das Merkmal der
Ähnlichkeit erfüllt?**



Wann ist das Merkmal der Ähnlichkeit erfüllt?

- Wie viel Übereinstimmung darf es geben ?

Wann ist das Merkmal der Ähnlichkeit erfüllt?

- Wie viel Übereinstimmung darf es geben ?
- Wie viel Abweichung darf es geben ?

Wann ist das Merkmal der Ähnlichkeit erfüllt?

- Wie viel Übereinstimmung darf es geben ?
- Wie viel Abweichung darf es geben ?
- Ob etwas ähnlich ist liegt im „Auge des Betrachters“

Beispiel: String-Vergleich in Java

```
String personOneName="Hanni",
      personTwoName="Nanni";

//Vergleich des Inhaltes der 2 Objekte

if(personOneName.equals(personTwoName))
    System.out.println("Die Namen sind gleich !");
else
    System.out.println("Die Namen sind ungleich !");
```

String-Vergleich in Java

- Die Methode *equals* prüft auf Gleichheit

Beispiel: String-Vergleich in Java

- Die Methode *equals* prüft auf Gleichheit
- Aussagen, in welchem Maße die Zeichenketten ähnlich sind, kann *equals* nicht liefern

Festlegen einer Schwelle

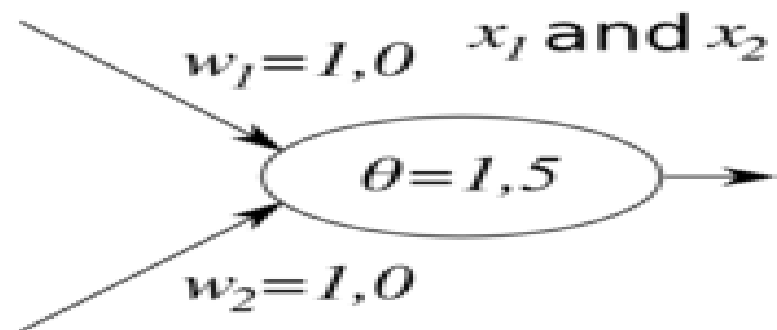
Um dennoch eine Aussage treffen zu können, in welchem Maße die Namen „Hanni“ und „Nanni“ gleich sind, muss eine Definition gefunden werden, die konkret festlegt, wann etwas ähnlich ist.

Beispiel: Schwelle eines künstlichen Neuron

Die Eingangssignale eines künstlichen Neurons werden aufsummiert und anschließend mit einem Schwellenwert verglichen. Ist die Summe der Eingangssignale größer als der Schwellenwert feuert das Neuron.

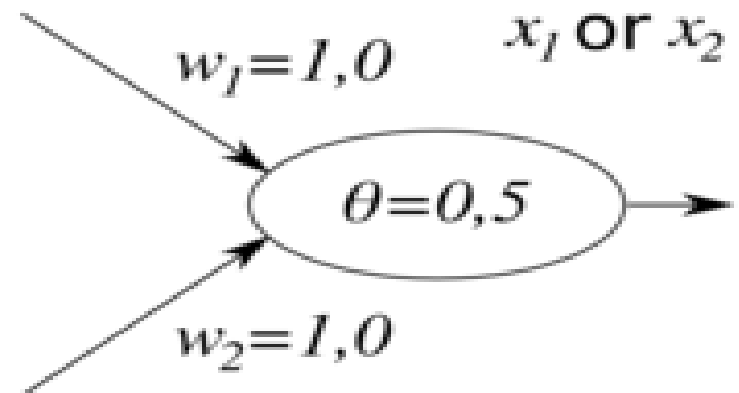
Beispiel: Und

A	B	$X = A * B$
0	0	0
0	1	0
1	0	0
1	1	1



Beispiel: Oder

A	B	$X = A+B$
0	0	0
0	1	1
1	0	1
1	1	1



Sequenz-Alignments in Dotplot



NeoBio-API

- Bioinformatik Algorithmen in Java

NeoBio-API

- Bioinformatik Algorithmen in Java
- Entwickelt von King's College London

NeoBio-API

- Bioinformatik Algorithmen in Java
- Entwickelt von King's College London
- Lizenz: GNU General Public License

NeoBio-API

- Bioinformatik Algorithmen in Java
- Entwickelt von King's College London
- Lizenz: GNU General Public License
- Alignment-Algorithmen wurde mit Mitteln der dynamischen Programmierung erstellt

Exkurs: Dynamische Programmierung

- Die optimale Lösung für ein Problem der Größe n setzt sich aus optimalen Teillösungen kleinerer Größe zusammen
(Bellmannsches Optimalitätsprinzip)

Exkurs: Dynamische Programmierung

- Die optimale Lösung für ein Problem der Größe n setzt sich aus optimalen Teillösungen kleinerer Größe zusammen (Bellmannsches Optimalitätsprinzip)
- Umfangreiche Probleme werden in viele kleine Probleme zerlegt, die unabhängig voneinander gelöst werden können

Exkurs: Dynamische Programmierung

- Die optimale Lösung für ein Problem der Größe n setzt sich aus optimalen Teillösungen kleinerer Größe zusammen (Bellmannsches Optimalitätsprinzip)
- Umfangreiche Probleme werden in viele kleine Probleme zerlegt, die unabhängig voneinander gelöst werden können
- Prinzip: „Teile und Herrsche“

Übersicht der verwendeten Alignment-Algorithmen in Dotplot

- Needleman-Wunsch-Algorithmus
- Smith-Waterman-Algorithmus
- Crochemore-Landau-ZivUkelson-Algorithmus

Needleman-Wunsch-Algorithmus

- Berechnung von globalen Alignments mit Hilfe von dynamischer Programmierung

Needleman-Wunsch-Algorithmus

- Berechnung von globalen Alignments mit Hilfe von dynamischer Programmierung
- Zeichenketten sollten ähnliche Längen aufweisen

Needleman-Wunsch-Algorithmus

- Berechnung von globalen Alignments mit Hilfe von dynamischer Programmierung
- Zeichenketten sollten ähnliche Längen aufweisen
- Gaps sind erlaubt

Needleman-Wunsch-Algorithmus

- Berechnung von globalen Alignments mit Hilfe von dynamischer Programmierung
- Zeichenketten sollten ähnliche Längen aufweisen
- Gaps sind erlaubt
- Problem: es ist nicht möglich die längste übereinstimmende Teilsequenz beider Sequenzen zu finden

Smith-Waterman-Algorithmus

- Modifikation des Needleman-Wunsch-Algorithmus

Smith-Waterman-Algorithmus

- Modifikation des Needleman-Wunsch-Algorithmus
- Berechnet das optimale lokale Alignment unter der Verwendung der dynamischen Programmierung.

Crochemore-Landau-ZivUkelson-Algorithmus

Es existieren zwei Implementierungen des Crochemore-Landau-ZivUkelson-Algorithmus. Jeweils eine Implementierung für globale Alignments, sowie eine für lokale Alignments.

Problem der Algorithmen

- NeoBio-Algorithmen sind für DNA-Sequenzen konzipiert

Problem der Algorithmen

- NeoBio-Algorithmen sind für DNA-Sequenzen konzipiert
- Eine DNA-Sequenz ist eine Folge von Buchstaben

Problem der Algorithmen

- NeoBio-Algorithmen sind für DNA-Sequenzen konzipiert
- Eine DNA-Sequenz ist eine Folge von Buchstaben
- Zahlen und Sonderzeichen werden nicht unterstützt

Problem der Algorithmen

- NeoBio-Algorithmen sind für DNA-Sequenzen konzipiert
- Eine DNA-Sequenz ist eine Folge von Buchstaben
- Zahlen und Sonderzeichen werden nicht unterstützt

Problem der Algorithmen

Eingabe:

a="gap1"

b="gpa2"

Problem der Algorithmen

Eingabe:

a="gap1"

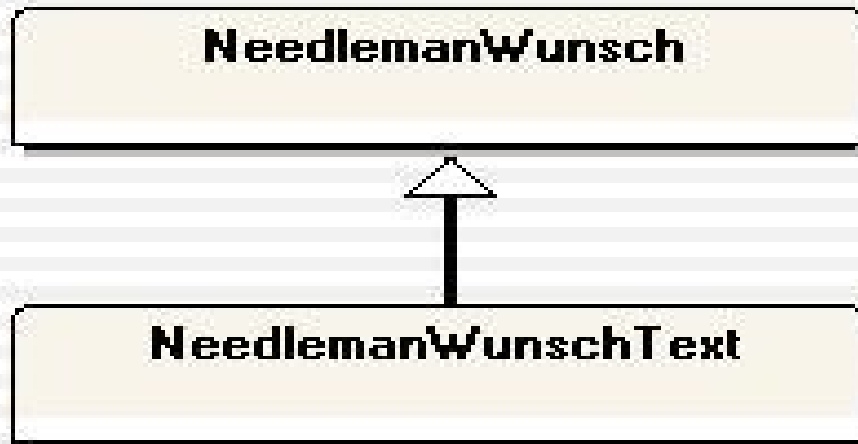
b="gpa2"

Ausgabe:

```
InvalidSequenceException: Sequences can contain letters only.  
  at CharSequence.<init>(CharSequence.java:73)  
  at ComparatorTest.test(ComparatorTest.java:51)  
  at ComparatorTest.main(ComparatorTest.java:8)
```

Erweiterung der Algorithmen

`::org.dotplot.alignments.neobio`

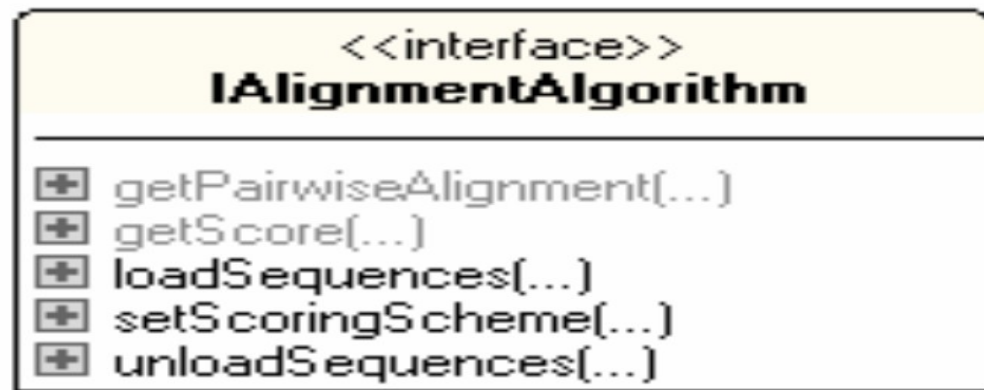


Schnittstelle für Sequenz-Alignments

A light green square is in the top-left corner. A dark blue horizontal bar with rounded left ends spans the width of the slide, positioned below the title.

Schnittstelle für Sequenz-Alignments

`::org.dotplot.alignments`



<<interface>>

org.dotplot.alignments::IAlignmentAlgorithm

- + getPairwiseAlignment(...)
- + getScore(...)
- + loadSequences(...)
- + setScoringScheme(...)
- + unloadSequences(...)

PairwiseAlignmentAlgorithm

- + getPairwiseAlignment(...)
- + getScore(...)
- + loadSequences(...)
- + setScoringScheme(...)
- + unloadSequences(...)

NeedlemanWunsch

- + getId(...)
- + getInfo(...)
- + getName(...)
- + setId(...)
- + setInfo(...)
- + setName(...)



Interface `java.util.Comparator`

- Die Schnittstelle besitzt 2 Methoden:
 - `public boolean equals(Object o)`
 - `public int compare(Object o1, Object o2)`

Interface `java.util.Comparator`

- Die Schnittstelle besitzt 2 Methoden:
 - `public boolean equals(Object o)`
 - `public int compare(Object o1, Object o2)`



Interface `java.util.Comparator`

- Die Schnittstelle besitzt 2 Methoden:
 - `public boolean equals(Object o)`
 - `public int compare(Object o1, Object o2)`



Vorsicht !

Die Methode „compare“ vergleicht die beiden übergebenen Objekte und liefert eine negative Zahl, Null, oder eine positive Zahl, je nachdem, ob o1 im Sinne der Ordnung kleiner, gleich oder größer als o2 ist.

AlignmentComparator Anforderungen

- Einfache Übergabe der zu vergleichenden Sequenzen

AlignmentComparator Anforderungen

- Einfache Übergabe der zu vergleichenden Sequenzen
- Übergabe einer benutzerdefiniert Schwelle

AlignmentComparator Anforderungen

- Einfache Übergabe der zu vergleichenden Sequenzen
- Übergabe einer benutzerdefiniert Schwelle
- Mögliches Ergebnis sollte nur „ist ähnlich“ oder „ist nicht ähnlich“ sein






Implementierung eines AlignmentComparator



Implementierung eines AlignmentComparator

`::org.dotplot.alignments`

AlignmentComparator

-  `AlignmentComparator(...)`
-  `compare(...)`
-  `getAlignmentAlgorithm(...)`
-  `getBasicScoringScheme(...)`
-  `getName(...)`
-  `getThreshold(...)`
-  `init(...)`
-  `setAlignmentAlgorithm(...)`
-  `setBasicScoringScheme(...)`
-  `setBasicScoringScheme(...)`
-  `setName(...)`
-  `setThreshold(...)`

Die *compare*-Methode

```
public int compare(String token1, String token2) {
    int score=0;
    try {
        alignmentAlg.loadSequences(new StringReader(token1),new StringReader(token2));
        alignmentAlg.setScoringScheme(bss);
        PairwiseAlignment pa= alignmentAlg.getPairwiseAlignment();
        score=pa.getScore();
    }
    catch(Exception exc) {
        exc.printStackTrace();
    }

    int perfectMatch=(token1).length() * match_reward;
    if(perfectMatch - score >= threshold)
        return 0; //sind ungleich
    else
        return 1; //sind gleich
}
```

Beispiel: String-Vergleich mit AlignmentComparator

```
int match=1,
    mismatch=-1,
    gap=-2;
AlignmentComparator comparator=new AlignmentComparator();

comparator.setAlignmentAlgorithm(new SmithWatermanText());
comparator.setBasicScoringScheme(match,mismatch,gap);
comparator.setThreshold(1);

if( comparator.compare("Hanni","Nanni") == 0 )
    System.out.println("Die Namen sind ähnlich.");
else
    System.out.println("Die Namen sind nicht ähnlich.");
```

Fazit

- Sequenz-Alignments bieten eine gute Alternative um „unscharfe“ Aussagen zu treffen
 - Aussagen wie „ist ähnlich“ bzw. „ist nicht ähnlich“ können getroffen werden

Fazit

- Sequenz-Alignments bieten eine gute Alternative um „unscharfe“ Aussagen zu treffen
 - Aussagen wie „ist ähnlich“ bzw. „ist nicht ähnlich“ können getroffen werden
- Viele Implementierungen sind frei verfügbar

Fazit

- Sequenz-Alignments bieten eine gute Alternative um „unscharfe“ Aussagen zu treffen
 - Aussagen wie „ist ähnlich“ bzw. „ist nicht ähnlich“ können getroffen werden
- Viele Implementierungen sind frei verfügbar
- Leider sind viele Implementierungen speziell für DNA-Analysen entwickelt

Quellen

- *„Sequenz – Alignment“
(von Dr. Rainer König)*
- *„NeoBio - Bioinformatics Algorithms in Java“
(www.neobio.sourceforge.org)*
- *„Sequenzalignment“
(www.wikipedia.org)*

Fragen ?

